A METHOD OF LEARNING RULES FROM UNCERTAIN DATA APPLIED TO THE COMPUTER VISION PROBLEM

D. Hutber and P. Sims
Sowerby Research Centre (FPC 267)
British Aerospace PLC, Naval Weapons Division
P.O.Box 5, Filton, Bristol BS12 7QW, England

ABSTRACT

The construction of knowledge-based systems of a size large enough to be useful has led to problems of knowledge acquisition. A way of solving this is to enable the computer to automatically generate its own knowledge from sets of sample data. This becomes further complicated when the sample data may have errors or noise in it.

This paper describes a system that generates knowledge in the form of rules from uncertain data, in the domain of computer vision. The way in which the uncertainty arises and is processed is discussed, and some sample results are presented.

KEYWORDS: machine learning, fuzzy sets, computer vision.

INTRODUCTION

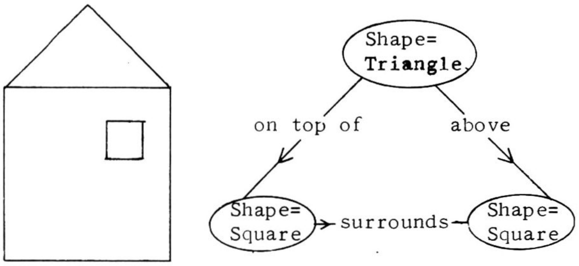
The construction of knowledge-based systems of a size large enough to be useful has led to problems of data acquisition. Expert systems have relied on the interaction between a knowledge engineer and a domain expert to produce a set of rules that capture the expert's knowledge on a particular topic. This process is very time-consuming, and as the size of the knowledge base increases it becomes a limiting factor. In computer vision, this method has the additional problem that the language necessary to represent the rules is not well-defined. The information given to any vision system is usually in the form of pixels, but formulating rules in terms of pixels is computationally expensive and would be difficult for a programmer to understand. A higher-level representation language is required in order to bring down the computation cost and to aid comprehension.

This paper addresses the subject of 'machine learning from examples', or equivalently of automatically generating rules to describe a concept from examples and counter-examples of that concept. Desirable properties of such a generation process are ease of inclusion of additional problem-specific knowledge, and ease of comprehension by a user or programmer. The representation of the examples and rules is hence of primary importance, since to a large extent this will determine the range of situations that can be expressed, and the manipulations it is possible to perform.

The problem of interpreting uncertain data has received considerable attention from people building expert systems, e.g. MYCIN (Shortliffe Buchanan 1), but the problem of learning rules to describe uncertain data has been studied less. In computer vision there is uncertainty due to imperfect image processing and noise. Here this has been modelled by the technique of fuzzy sets.

EXAMPLES AND COUNTER-EXAMPLES

Objects are made up of sub-objects called 'primitives'. The primitives have properties that are called 'attributes', and there are connections between the primitives which are expressed as relations. For the purposes of computer vision, these primitives are the regions, and the attributes may be properties such as shape, size and colour; the relations are 2-D spatial relationships such as 'above' or 'surrounds'. This representation corresponds to a semantic net or graph.



Here shape, size and height are the unary descriptors used and these take values of, for example, shape=triangle, size=medium and height=6. This illustrates the use of two types of unary descriptors:

- nominal descriptors, where the values have names.
- linear descriptors, where the values are numbers.

The two types of unary descriptor are treated in different ways. More restrictions are placed on linear descriptors since it is assumed that they are ordered in a meaningful way, and that integer values differing by a small amount give rise to similar properties and can be grouped together.

Nominal descriptors have discrete names with no ordering implied on them. For example, it is not meaningful to describe a shape as halfway between a circle and a square, although as will be shown later, it is possible to express equal uncertainty as to whether a primitive's shape is 'circle' or 'square'.

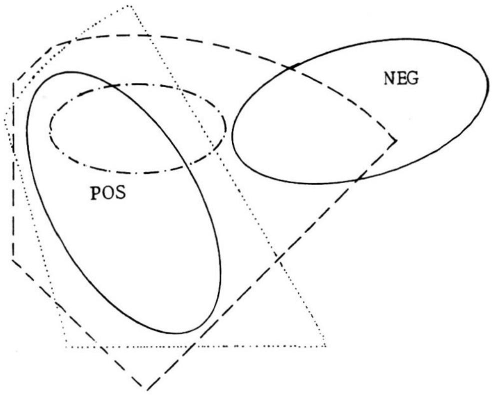
The only binary descriptor used at present is 'is spatially related to'. This takes values of, for example, 'surrounds' and is a nominal descriptor. Each value has an inverse, e.g., 'is-surrounded-by'.

GENERAL METHOD

The central idea in the learning algorithm described here is one of 'generate and test'. The method is an extension of the INDUCE algorithm (Michalski 2) where a series of trial descriptions is generated using a 'seed' example, and tested against examples and counter-examples. The seed example provides the descriptors from which the trial descriptions are constructed. After a description has been tested, if it is ranked better than those before it in the series, according to some criterion, it is retained and used to produce several more descriptions. The new descriptions are produced by specialising the old description. If it is no better than those before it, the trial description is discarded.

This guided generation process is equivalent to a search. The search is over a space of all possible descriptions, consisting of properties of, and relations between sub-objects, and this is guided by a set of examples of a concept and a set of counter-examples. These representatives are very important to the working of the algorithm, and so a good choice of examples, and more critically counter-examples, is essential (Winston 3).

In simple set terms, if we consider a space of all possible objects, then we may represent POS, the set of examples, and NEG, the set of counter-examples, by the sets shown in the figure below:



A trial description will cover a number of possible objects. Three descriptions are represented by the sets in dashed lines, each of which has a different property. The set----covers all the examples and is called a 'complete description', and the set -·-· covers none of the counter-examples and is called a 'consistent description'. The aim of the learning algorithm is to produce a number of complete and consistent descriptions, for example, the set ...., which can then be used on unknown objects to identify them as members of the concept.

The difference between this type of learning and a decision tree is the inductive process, whereby descriptions of a class of objects are induced from the sets of examples and counter-examples. The induction in this case is performed by generalisation rules which act on a description to produce a more general description. In terms of the set diagram above, generalisation increases the range of objects that the description covers.

GENERALISATION RULES

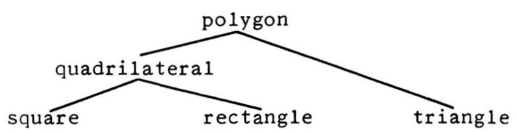
Generalisation is performed only on consistent descriptions generated by the learning algorithm (i.e. those descriptions which do not cover any of the counter-examples). The reason for this is that the aim of the algorithm is to produce a series of consistent descriptions that are as simple as possible. Hence when a consistent (but not complete) description is produced, it is generalised, hoping that the new description will cover more examples in POS whilst maintaining the consistency property.

There are really only two generalisation rules used in the implementation, and they correspond to internal disjunction of values of the two types of descriptor. They are:

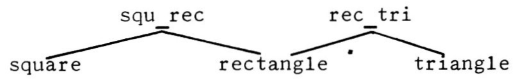
(i) Adding alternative (or range of alternatives).
(ii) Closing interval.

(Only the first of these will be described in detail.)

(i) The adding alternative rule works on nominal descriptors, using two values of the same descriptor, one of which is in the description already, and the other which it is desired to include. Values of the descriptor may already be grouped together, either by the programmer or by the system when it has learnt a rule in the past, to form a structure. The existing structure of the descriptor is searched to find all existing groupings of values that include both of the required values. For example, the 'shape' descriptor may have the structure shown below:



Here 'quadrilateral' is defined as 'square or rectangle' and 'polygon' as 'quadrilateral or triangle' by the user of the program. The structure is in this case a tree which can be used as a convenient means of generalisation, so that square is-a-kind-of quadrilateral is-a-kind-of polygon. However, this restricted generalisation has the disadvantage that the user has to supply all the structure, and that if the combination 'rectangle or triangle' (but not square) appeared repeatedly this could not be expressed efficiently. An alternative structuring technique adopted by this work is a 'group if useful' technique, where initially the user can specify as much or as little grouping as he sees fit, and the program will group together values if it repeatedly finds such a process useful. Thus if no structure were supplied to the 'shape' descriptor, but the combinations 'square or rectangle' and 'rectangle or triangle' occurred frequently as the algorithm ran, then the structure of the descriptor would look like the figure below:



The language being used here is clearly less comprehensible than that used in the previous structure, but for display purposes the original 'square or rectangle', etc. may be used. It has the advantages of being easier to manipulate and being able to express a wider variety of combinations of values.

Each of the groupings containing the two values is used to form a generalised description which is tested for consistency, starting with the largest grouping (corresponding to the most general description) and going on to the smallest. When a consistent generalised description is found the process stops. It is only required to test this series of descriptions on the counter-examples to establish the consistency property. If no consistent generalised description is found, a group consisting of the two values only is created and the corresponding description is tested. If this fails, the two-value group is deleted and generalisation of this descriptor is abandoned.

(ii) The closing interval rule works in much the same way on linear descriptors, using intervals including the two values rather than on groupings.

STRUCTURE AND ATTRIBUTES

The way that the algorithm is implemented is to process the binary descriptors first, generating structure-only descriptions. Each of these structure descriptions is then used as a framework in which to run the algorithm for the unary descriptors. There are two effects arising from the separation of the unary and binary descriptors. These are:

(1) Since structure is treated first, the algorithm preferentially generates solutions with structural conditions rather than attribute conditions.

(2) Any description obtained with a consistent structural part will be consistent.

PREFERENCE CRITERIA

The progression of the learning algorithm is influenced by two separate preference criteria. These are now described, and their effect on the type of description generated outlined.

Preference Criterion (1):

This is used on every description in order to quantify how close to being a solution it is. The measure used is simply:

Number of examples in POS covered by description -
number of counter-examples in NEG covered by description.

Preference Criterion (2):

The preference criterion used here is a cost function which states how successfully a description satisfies certain requirements. It is evaluated as a weighted sum of the length, cost, and degree of generalisation of a description. These weights are user defined according to the type of description it is wished to generate (e.g. long and specific or short and general). The contribution from each characteristic will be represented by a number between 0 and 1, defined as follows:

(i) Length of description
(ii) Cost of generating description
(iii) Degree of generalisation of description

The features used in this preference criterion are not exhaustive; for example, in a more complex system, computational simplicity, least possible memory used in storage and overall comprehensibility may be important characteristics for a description to exhibit.

UNCERTAINTY

The main difference between this system and those previously implemented is the way the quality of data relating to examples is treated. For example, a square might be a perfect example of a certain concept, but due to the imaging system it may not actually have a representation that exactly satisfies the axiomatic requirements for a square. Nevertheless it may have a certain perceptual similarity to a square, and may well be one in the actual scene which has become distorted in the imaging system.

There are several alternatives for representing uncertainty. In the majority of systems, Bayesian Probabilities have been favoured; however, Fuzzy Sets and the Shafer-Dempster approach (4) have also received attention in recent literature. Fuzzy sets (Zadeh 5) were selected to represent the uncertainty in the system.

Each fact is assigned a Fuzzy Truth Value (FTV) from 0 to 1. This value represents the degree of membership of the fact in the fuzzy set of true facts. Hence, a description which matches a series of facts from an example or counter-example will have a list of FTVs associated with it. These are then combined to give an overall fuzzy truth value for the description. If the description is made up of n descriptors, and the jth descriptor matches a fact in a specific Example with FTVψ_j then the FTV of the entire description (or the Degree of Fit of a description to an Example, E) is defined as:

$$F(E) = 0.5 + 1/n \left\{ \sum_{\psi_j > 0.5} \frac{(\psi_j - 0.5)^2}{0.5} - \sum_{\psi_j \leqslant 0.5} (0.5 - \psi_j) \right\} \quad (1)$$

The function F(E) has the following properties:

(i) Simple polynomial form.
(ii) Sensitive to all truth values (unlike MAX or MIN).
(iii) Independent of order of truth values.
(iv) Facts with FTVs<0.5 aregiven greater weight in the calculation than those with FTVs>0.5 .
(v) $F(E) < \psi_j$ for $n=1$, $0.5 < \psi_j < 1$.

EFFECTS OF THE INTRODUCTION OF UNCERTAINTY

The definitions of Consistency and Completeness now become dependent on the degree of fit. A consistency threshold Tconsistent is set such that a description will not be consistent if F(CE)≫Tconsistent for any counter-example CE. A completeness threshold Tcomplete is also set. If F(E)≫Tcomplete for an example E then that example is defined to be covered by that description.

The introduction of uncertainty into the definitions of Consistency and Completeness affects the evaluation of the Preference Criteria. With Preference Criterion (1) the definition is unchanged except that the number of examples in POS covered by the description will be those examples with degree of fit greater than the completeness threshold. Similarly, the examples covered in NEG covered by a description will be those with degree of fit greater than the consistency threshold.

Preference Criterion (2) is affected by the introduction of two new factors: the consistency and completeness ratings of a description, defined as follows:

(i) Consistency of description

If the consistency threshold is exceeded by any counter-example then the consistency condition is broken and the consistency rating is set to zero. If the degree of fit for the ith counter-example is F(CEi) (i=1..n) then:

$$\text{Consistency Rating} = 1 - \frac{1}{n} \sum_i \frac{F(CEi)}{Tconsistent} \quad (2)$$

(N.B. If F(CEi)=0 for all i
then Consistency Rating=1)

(ii) Completeness of description

If the degree of fit for the ith example is F(Ei) (i=1..n) then:

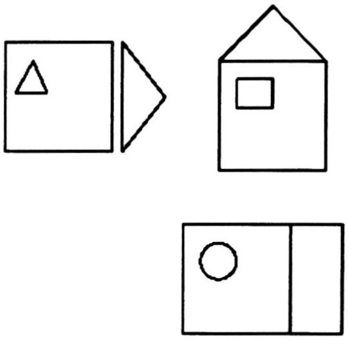
$$\text{Completeness Rating} = \frac{1}{n} \sum_{F(Ei) \gg Tcomplete} \frac{F(Ei)}{1} \quad (3)$$

(N.B. If F(Ei)=1 for all i
then Completeness Rating=1)

Hence, the evaluation of the Preference Criterion now becomes a weighted sum of five features: length, cost, degree of generalisation, consistency, completeness. The introduction of Completeness and Consistency ratings has two effects in guiding the system. By weighting in favour of completeness the system can be biased to include all positive examples. By weighting in in favour of Consistency the system can be biased against including any counter-examples.

RESULTS

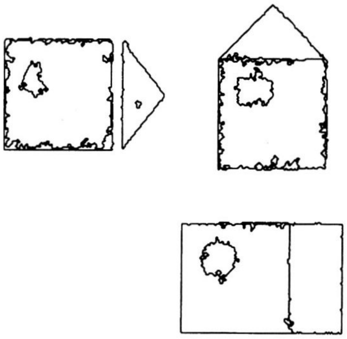
This section shows the results of running the algorithm on a synthetic image, before and after adding Gaussian noise to it. The different descriptions generated in each case are given:



Processed Version of Perfect Input Data.

Rule Generated:

'There are two objects X and Y such that
(X surrounds Y and
Y is a square or a circle)'



Processed Version of Imperfect Input Data.

Rule Generated:

'There are three objects X,Y and Z such that
(X surrounds Y and
Y is a rectangle) or
(X is right of Y and
X is right of Z and
X is a rectangle)'

The result of adding the noise is that the surrounded objects in the examples cannot be reliably labelled as a square and circle as before. The object in the top example is now considered more likely to be a rectangle and the surrounded object in the bottom example is too degraded to be incorporated as part of a rule. This results in the second half of the above rule being generated.

DISCUSSION

In its present form, the learning system described has several problems associated with it, due to the incorporation of uncertainty in the algorithm. Some of these problems are described below.

Coverage of Seed Example.

The INDUCE algorithm is guaranteed to give a solution and terminate eventually (when working on noise-free data), even if the rule obtained is a disjunctive list of the examples in POS. (In this case, no induction has been performed by the system.) The reason why the algorithm terminates is because all the descriptions generated using a 'seed' example are partial descriptions of the example and hence cover it. As the algorithm builds up longer partial descriptions of the seed example, the set of objects covered by the description become smaller, until eventually only the one example is covered.

The use of the degree of fit measure defined above means that partial descriptions of the example will not necessarily 'cover' (in the fuzzy sense) the seed chosen. A consequence of this is that the algorithm cannot be guaranteed to give a solution, unless some other constraints are placed on it. If the situation occurs in which the seed example may not be described without a counter-example also being covered, then to all intents and purposes the descriptors chosen do not discriminate between this example and the counter-example. This may be remedied in one of two ways:

(i) Alter the Tcomplete threshold to discover whether any setting will give discrimination.

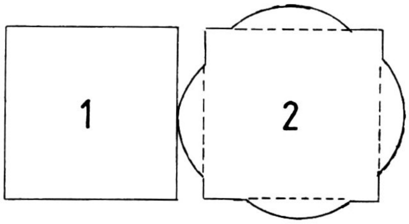
(ii) Use a better set of descriptors.

Degree of Fit Measure.

The degree of fit measure as defined in Equation 1 has the property that if two facts in an example with truth values of 0 and 1 respectively were matched to two descriptors making up a description, then that example would have a degree of fit of .5 to that description, in spite of the FTV of 0 which is designed to represent the falsity of that fact. This is resolved by making the additional assumption that if the degree of membership of a fact is less than a threshold value T (e.g. T=0.3), then it is deleted from the data base. This can prevent the matching of low membership facts and cut down the processing done by the algorithm.

Interdependence of Certainty Values.

This problem is perhaps best illustrated by an example. Consider the two primitives in the figure below:



If primitive 2 is a square, then it is not touching primitive 1.

If primitive 2 is a circle, then it is touching primitive 1.

In other words, the certainty of the relation '1 is touching 2' is dependent on the interpretation of the shape of primitive 1. It is therefore assumed for simplicity that the facts describing the examples are independent of each other, to an approximation.

CONCLUSIONS

In this work, a machine learning scheme for computer vision that models the effects of introducing uncertainty has been implemented. At present, this work is at an early stage and has only been applied to simple, synthetic image data to investigate the changes that occur when uncertainty is present. From the results obtained to date, it seems that the rules that are learnt from perfect data may differ significantly from those obtained from the imperfect equivalent.

ACKNOWLEDGEMENT

This research was supported in part by the Alvey Directorate as part of the Alvey project 'Identification of Objects in 2-D Images'.

REFERENCES

1. Shortliffe E.H. and Buchanan B.G. 'A Model of Inexact Reasoning in Medicine'. Math. Biosci., Vol. 23, 1985.

2. Michalski R.S., Carbonell J.G., Mitchell T.M. eds. 'Machine Learning: An Artificial Intelligence Approach'. Palo Alto: Tioga, 1983.

3. Winston P.H. 'Artificial Intelligence'. Addison-Wesley, 1984.

4. Shafer G. 'A Mathematical Theory of Evidence'. Princton University Press, 1976.

5. Zadeh L.A. 'Fuzzy Sets'. Information and Control, Vol. 8, pp 338-353, 1965.